

Рисование / Анимация



JavaScript
Courses

courses.dp.ua

Разбор ДЗ #1.2

Домашнее задание #J.2 Адресная книга v.2

Адресная книга

Поиск

Имя



Телефон

Адрес

Иван

+380675558823

London, United Kingdom

Используйте заготовку:
[./source/example_1](#)

Выведем в подготовленную разметку данные пользователей полученные от сервиса <https://randomuser.me/> Обеспечьте сортировку по имени. После загрузки список выводится в неотсортированном виде. Сортировка начинается только после нажатия кнопки. Обеспечьте поиск по списку. **Поиск и сортировка должны работать совместно!**

Рисование / Canvas

JavaScript и рисование



В основе рисования на веб-странице лежит тег **canvas**

Рисование, Графика, Canvas

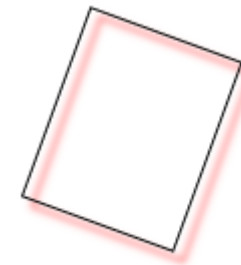
```
31 <canvas id="paint-canvas"></canvas>
32 <script>
33
34     var canvas      = document.getElementById("paint-canvas");
35     canvas.width    = canvas.clientWidth;
36     canvas.height   = canvas.clientHeight;
37
38     var context     = canvas.getContext("2d");
39
40     context.moveTo(200, 200);
41     context.lineTo(300, 250);
42     context.lineTo(200, 300);
43     context.closePath();
44     context.stroke();
45
46 </script>
```

*Тег **canvas** – представляет собой «холст», прямоугольную область в которой можно рисовать. Контекст canvas'a – объект который содержит множество методов для рисования на «холсте».*

Рисование, Графика, Canvas

*Рисование на **canvas**'е основано на отрисовке примитивов.*

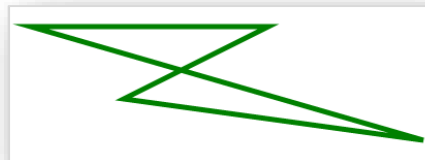
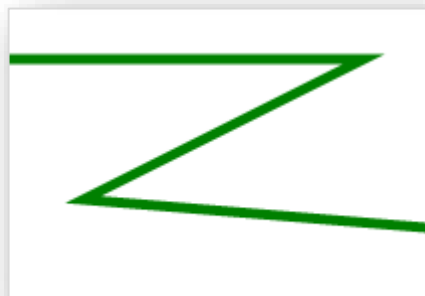
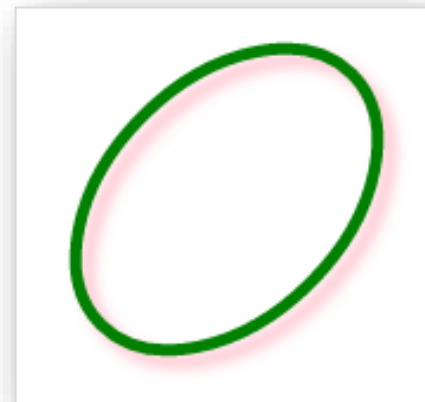
- 1) Штриховых (контурных фигур) – в названии методов и свойств есть слово **stroke**;*
- 2) Заполненных фигур, в названии методов и свойств есть слово **fill**;*
- 3) Наложении спецэффектов (тени, развороты, искажения и т.п.).*



Рисование, Графика, Canvas

Примитивы можно рисовать при помощи функций-заготовок: прямоугольник (**rect()**), эллипс (**ellipse()**) и т.п.

Либо самостоятельно задав контур фигуры состоящей из множества линий. Для этого есть функции **beginPath()** и **closePath()** – для случаев когда нужно замкнуть контур (между первой и последней точкой фигуры).



Рисование примитивов

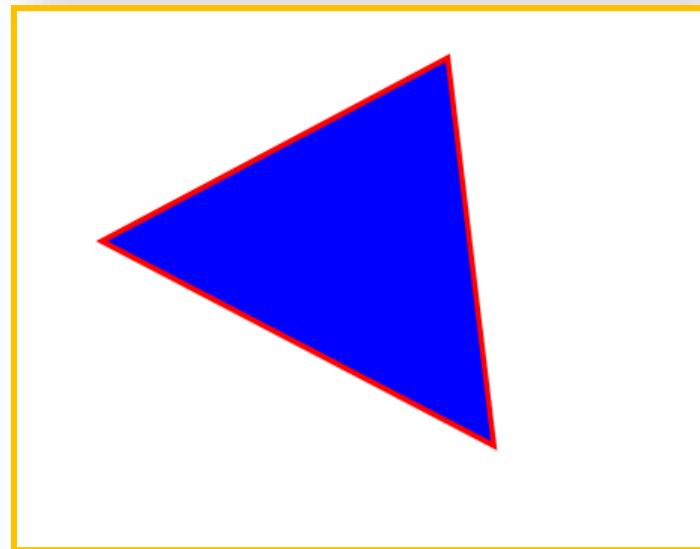
```
context.beginPath();  
context.moveTo(175, 225);  
context.lineTo(400, 113);  
context.lineTo(430, 350);  
//context.closePath();  
context.stroke();
```



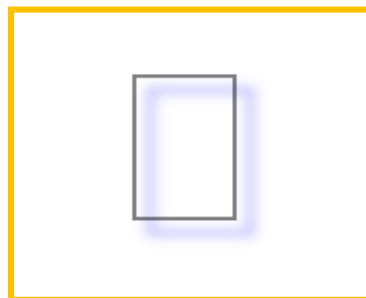
```
context.beginPath();  
context.moveTo(175, 225);  
context.lineTo(400, 113);  
context.lineTo(430, 350);  
//context.closePath();  
context.fill();
```



```
context.beginPath();  
context.moveTo(175, 225);  
context.lineTo(400, 113);  
context.lineTo(430, 350);  
context.closePath();  
context.lineWidth = 7;  
context.strokeStyle = "red";  
context.fillStyle = "blue";  
context.stroke();  
context.fill();
```



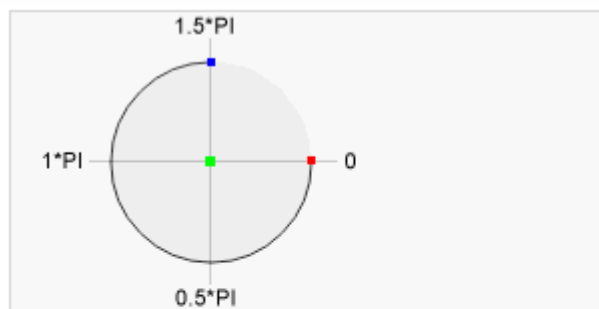
```
context.rect(300, 200, 50, 80);  
context.shadowBlur = 10;  
context.shadowOffsetY = 8;  
context.shadowOffsetX = 8;  
context.shadowColor = "blue";  
context.stroke();
```



Рисование примитивов

```
context.strokeStyle = "green";  
context.rect(300, 200, 50, 80);  
//rect(x,y, width, height)  
context.stroke();
```

```
context.arc(200, 300, 45, 1.8*Math.PI, 2*Math.PI, false);  
//arc(x, y, radius, startAngle, finishAngle, direction);  
context.stroke();
```



Center	<code>arc(100,75,50,0*Math.PI,1.5*Math.PI)</code>
Start angle	<code>arc(100,75,50,0,1.5*Math.PI)</code>
End angle	<code>arc(100,75,50,0*Math.PI,1.5*Math.PI)</code>

Прямоугольник и дуга

Свойства (графические атрибуты «холста»)

Paths

Method	Description
fill()	Fills the current drawing (path)
stroke()	Actually draws the path you have defined
beginPath()	Begins a path, or resets the current path
moveTo()	Moves the path to the specified point in the canvas, without creating a line
closePath()	Creates a path from the current point back to the starting point
lineTo()	Adds a new point and creates a line to that point from the last specified point in the canvas
clip()	Clips a region of any shape and size from the original canvas
quadraticCurveTo()	Creates a quadratic Bézier curve
bezierCurveTo()	Creates a cubic Bézier curve
arc()	Creates an arc/curve (used to create circles, or parts of circles)
arcTo()	Creates an arc/curve between two tangents
isPointInPath()	Returns true if the specified point is in the current path, otherwise false

Transformations

Method	Description
scale()	Scales the current drawing bigger or smaller
rotate()	Rotates the current drawing
translate()	Remaps the (0,0) position on the canvas
transform()	Replaces the current transformation matrix for the drawing

Подробнее: http://www.w3schools.com/tags/ref_canvas.asp

Paint на JavaScript

```

31 <canvas id="paint-canvas"></canvas>
32 <script>
33
34     var canvas      = document.getElementById("paint-canvas");
35     canvas.width    = canvas.clientWidth;
36     canvas.height   = canvas.clientHeight;
37
38     var context      = canvas.getContext("2d");
39     context.lineJoin  = "round";
40     context.lineWidth = 10;
41     context.strokeStyle = "red";
42
43     var x;
44     var y;
45     var painting = false;
46
47     canvas.onmousedown = function(e) {
48         painting = true;
49         x = e.offsetX;
50         y = e.offsetY;
51         context.moveTo(x, y);
52     };
53
54     canvas.onmousemove = function(e) {
55         if(painting) {
56             context.lineTo(e.offsetX, e.offsetY);
57             context.closePath();
58             context.stroke();
59             x = e.offsetX;
60             y = e.offsetY;
61             context.moveTo(x, y);
62         }
63     }
64
65     canvas.onmouseup = canvas.onmouseleave = function(e) {
66         painting = false;
67     };
68
69 </script>

```

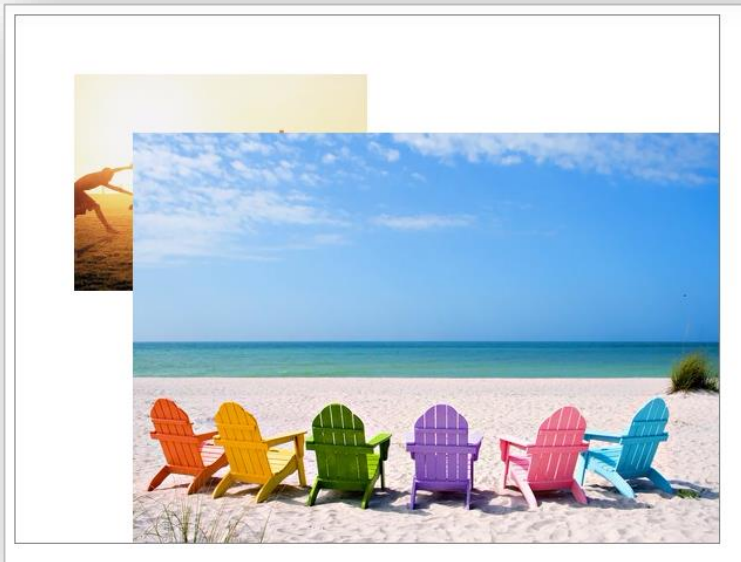
Paint на JavaScript

*Простой аналог
программы **paint** на базе
JavaScript и canvas.*

Отрисовка изображений

Загрузка изображений на холст (canvas)

```
var img1 = new Image();  
img1.src = "http://js.courses.dp.ua/files/15/1.jpg";  
img1.onload = function(){  
    context.drawImage(img1, 100, 100);  
}  
  
var img2 = new Image();  
img2.src = "http://js.courses.dp.ua/files/15/2.jpg";  
img2.onload = function(){  
    context.drawImage(img2, 50, 50, 250, 185);  
}
```



Загрузка изображений на холст / метод `drawImage()`

Метод `.drawImage()` выводит ранее загруженное изображение на «холст»

`drawImage(image, sx, sy, sWidth, sHeight, dx, dy, dWidth, dHeight)`

`image` — объект `Image()`;

`sx, sy` — координаты верхнего левого вставляемого изображения;

`sWidth, sHeight` — размеры вставляемого изображения;

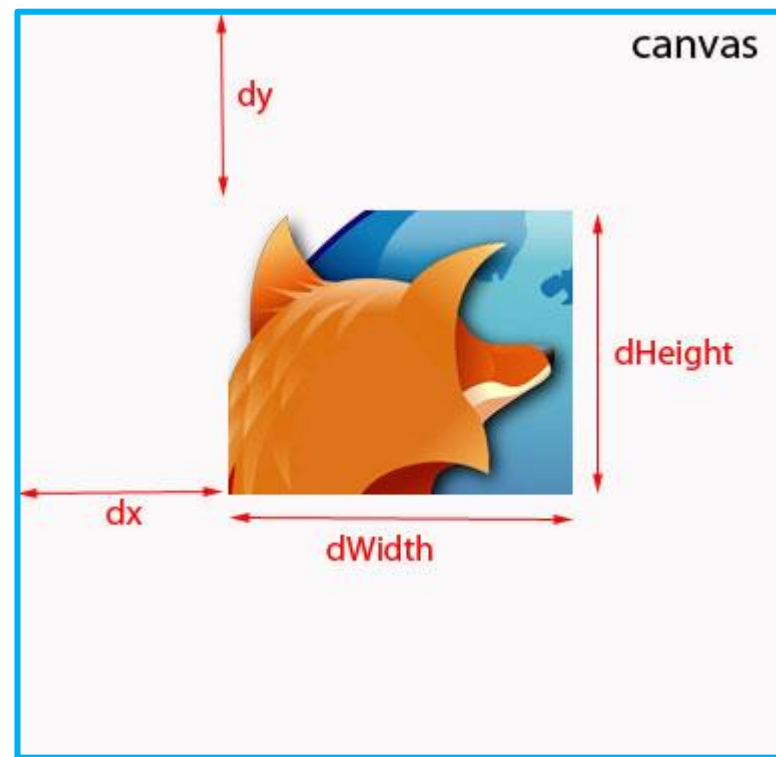
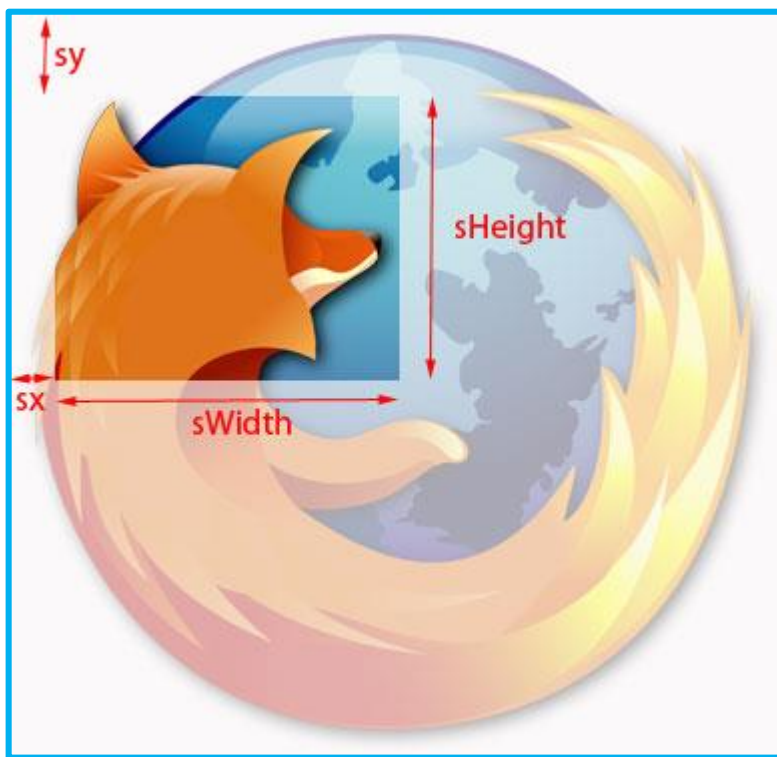
`dx, dy` — координаты размещения верхнего левого угла обрезанного изображения на холсте;

`dWidth, dHeight` — размеры обрезанного изображения на холсте.

Подробнее: https://www.w3schools.com/tags/canvas_drawimage.asp

Загрузка изображений на холст / метод `drawImage()`

Метод `.drawImage()` выводит ранее загруженное изображение на «холст»

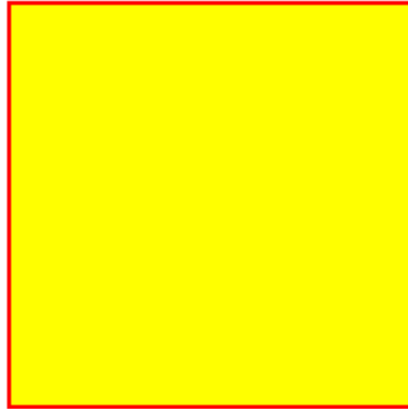


`drawImage(image, sx, sy, sWidth, sHeight, dx, dy, dWidth, dHeight)`

Анимация и JavaScript

*Анимация на веб-странице это,
как правило, изменение во времени
того или иного стилевого
свойства элемента во времени.*

***Немного практики: краткая история анимации
в браузере.***



Воспользуйтесь заготовкой: [./source/example_2](#)

Анимация при помощи setInterval()

```
14 <script>
15     window.onload = function() {
16         var offset = 0;
17         var step    = 3;
18
19         setInterval(function() {
20             tag.style.left = offset + "px";
21             tag.style.top  = offset + "px";
22
23             offset += step;
24
25             step = ((offset > 400) || (offset < 1)) ? -step : step;
26
27         }, 30);
28     }
29 </script>
```

Вроде и работает, но ресурсоёмко.

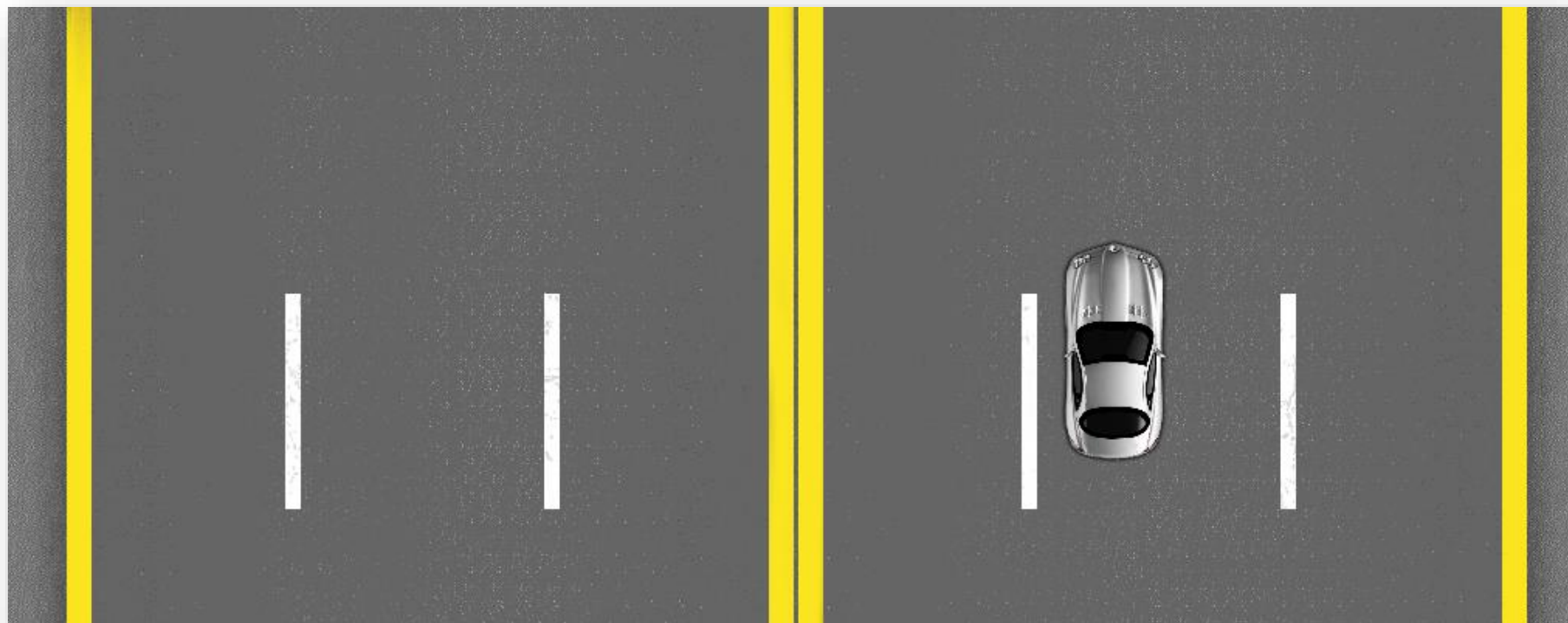
Анимация при помощи `requestAnimationFrame()` и FPS

```
14 <script>
15     window.onload = function() {
16         var offset = 0;
17         var step    = 3;
18
19         var fps = 15;
20
21         function move() {
22             setTimeout(function() {
23                 requestAnimationFrame(move);
24             }, 1000/fps);
25
26             tag.style.left = offset + "px";
27             tag.style.top  = offset + "px";
28
29             offset += step;
30
31             step = ((offset > 400) || (offset < 1)) ? -step : step;
32         }
33
34         move();
35     }
36 </script>
```

Уже лучше, теперь отрисовка происходит в те моменты когда браузер «готов» к перерисовке страницы. С помощью такой техники мы можем делать анимацию с желаемой частотой кадров.

Механика игровой анимации

Немного практики: механика игровой анимации.



Воспользуйтесь заготовкой: [./source/example_2](#)

Немного практики: механика игровой анимации

```
11 <script>
12     var context      = document.querySelector("canvas").getContext("2d");
13
14     var road_url      = "http://files.courses.dp.ua/js/21/road_2.jpg";
15     var road_image    = new Image();
16     road_image.src    = road_url;
17
18     var car_url       = "http://files.courses.dp.ua/js/21/car.png";
19     var car_image     = new Image();
20     car_image.src     = car_url;
21
22
23     var offset        = 1;
24     var step          = 3;
25     var car_x         = 200;
26     var car_y         = 200;
27
28     function go(){
29         offset += step;
30         if(offset + 511 > 1024) offset = 0;
31         requestAnimationFrame(go);
32         context.drawImage(road_image, 0, 511-offset, 1024, 400, 0, 0, 1024, 400);
33         context.drawImage(car_image, car_x, car_y, 90, 140);
34     }
35     requestAnimationFrame(go);
36
37     document.body.onkeydown = function(e){
38         if(e.code == "ArrowLeft")   car_x -= 4;
39         if(e.code == "ArrowRight")  car_x += 4;
40         if(e.code == "ArrowUp")     car_y -= 4;
41         if(e.code == "ArrowDown")   car_y += 4;
42         if(e.code == "ControlLeft") step = 10;
43         if(e.code == "Space")       step = 1;
44     }
45
46     document.body.onkeyup = function(e){
47         if(e.code == "ControlLeft") step = 2;
48         if(e.code == "Space")       step = 2;
49     }
50
51 </script>
```

*Игровая механика
сводится к перерисовке
на странице
изображений-заготовок
(спрайтов) на экране в
зависимости от
«ситуации» в игре.*

Домашнее задание
/узнать

О валидации данных формы средствами HTML



vUdav 10 июля 2017 в 17:53

Простая валидация формы без JS

CSS, HTML

Из песочницы



В данной статье я бы хотел поделиться методом быстрой валидации полей с помощью разметки и стилей. Данный метод не является кроссбраузерным и рекомендуется к использованию только как дополнительная фишка. По ходу статьи мы будем уменьшать наши шансы на кроссбраузерность, но повышать функциональность.

Давайте попробуем собрать стандартную форму, которая будет включать в себя: Имя, E-Mail, Телефон, Ссылку на сайт и допустим Ваш рост, чтобы поэкспериментировать с числовым полем.

<https://habr.com/ru/post/332804/>

О валидации данных формы средствами HTML (более подробно)

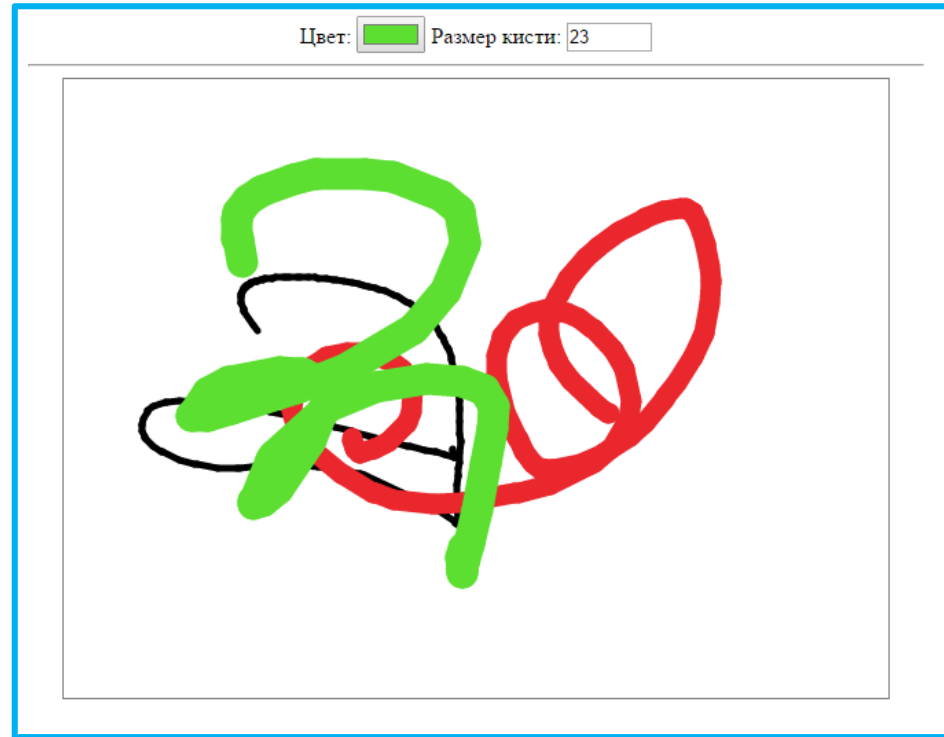
Проверка данных формы

Изучение веб-разработки › Изучение HTML: руководства и уроки › Руководство по HTML-формам › Проверка данных формы

https://developer.mozilla.org/ru/docs/Learn/HTML/Forms/Валидация_формы

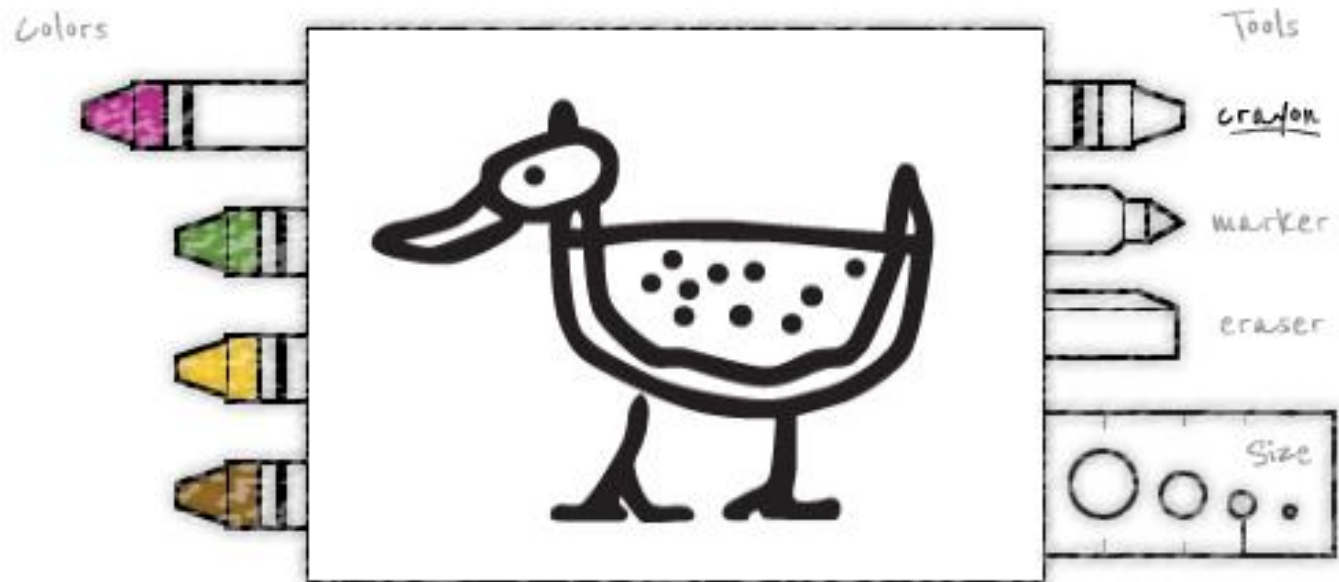
Домашнее задание
/сделать

Домашнее задание #К.1



Воспользуйтесь заготовкой: [./homework/index.html](http://homework/index.html) и сделайте Paint настраиваемым, т.е. необходимо сделать чтобы пользователь мог выбрать цвет линии и толщину линии. Например как в примере: [./homework/demo.html](http://homework/demo.html)

При решении ДЗ #К.1 вам поможет



<http://www.williammalone.com/articles/create-html5-canvas-javascript-drawing-app/>